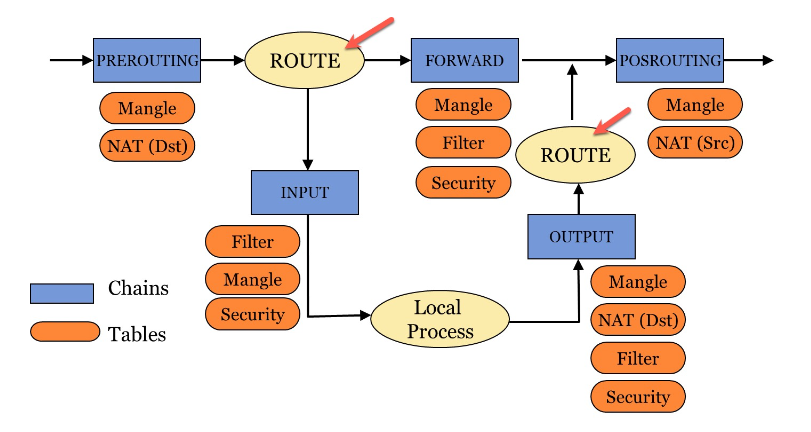
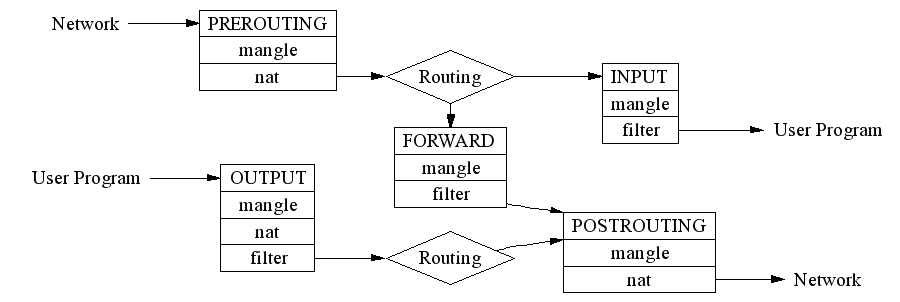
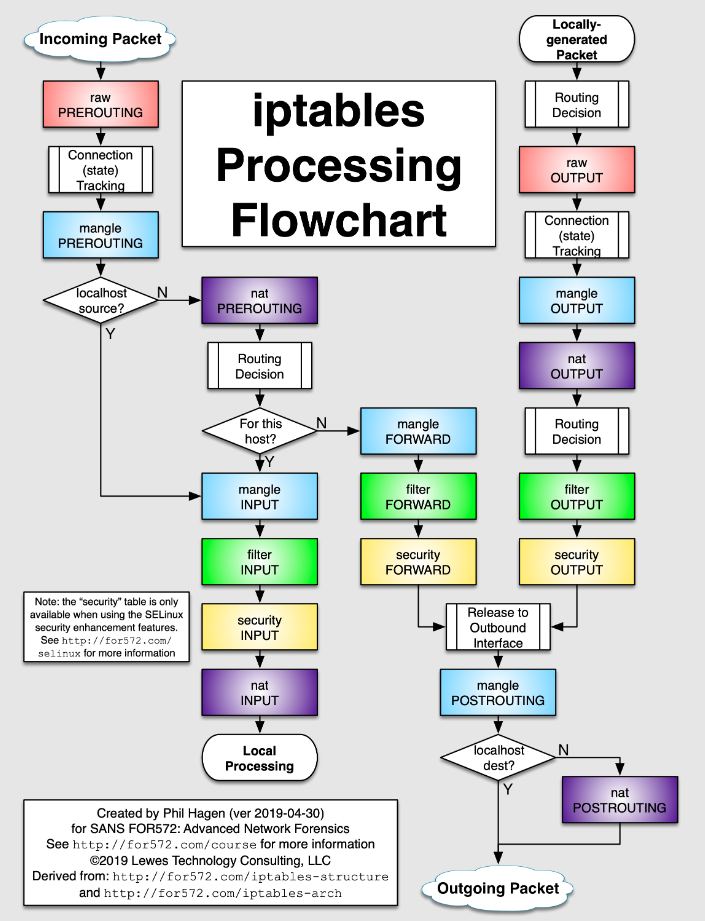
**Iptables**

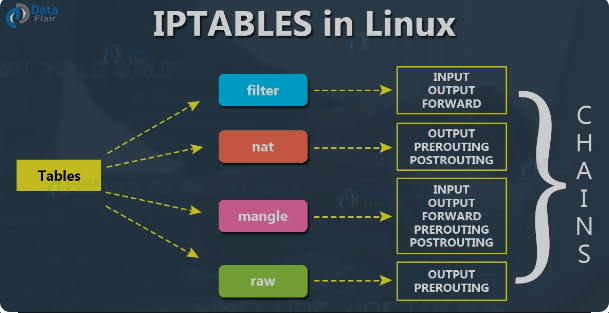
Traffic flow of a packet in linux through iptables is as below:







There are 3 default table in iptables and its chains are as below



# Raw table

This table includes raw traffic. It means before applying any changes packets go to raw tables. So in prerouting and output chains involve these table.

# Forward table

This table is the main table and decide which packet we want to accept or drop. As it is shown in pic, all packets those want to come into server or go out from server or forward from our server are dealt with this table. This table is involve input, output and forward chain.

# Nat table

Packets which must be source or destination NAT must go to this table. So it involves prerouting , postrouting and output chains.

# Mangle table

table is used to alter the IP headers of the packet in various ways. For instance, you can adjust the TTL. So we can see this table is involved in all chains.

# Security Table

table is used to set internal SELinux security context marks on packets, which will affect how SELinux or other systems that can interpret SELinux security contexts handle the packets.

# Chain Traversal Order

* **Incoming packets destined for the local system**: PREROUTING -> INPUT
* **Incoming packets destined to another host**: PREROUTING -> FORWARD -> POSTROUTING
* **Locally generated packets**: OUTPUT -> POSTROUTING

# IPTables Rules

Rules are placed within a specific chain of a specific table

## Matching

specifies the criteria that a packet must meet in order for the associated action (or “target”) to be executed. Rules can be constructed to match by

* protocol type
* destination address
* source address
* destination port
* source port
* destination network
* source network
* input or output interface
* headers
* connection state

## target

A “target” refers to the actions that are triggered when a packet meets the matching criteria of a rule

# IPTables and Connection Tracking

connection tracking system implemented on top of the netfilter framework. This functionality it needs to perform “stateful” operations.

## Available States

Connections tracked by the connection tracking system will be in one of the following states:

* NEW: When a packet arrives that is not associated with an existing connection, but is not invalid as a first packet, a new connection will be added to the system with this label. This happens for both connection-aware protocols like TCP and for connectionless protocols like UDP.
* ESTABLISHED: A connection is changed from NEW to ESTABLISHED when it receives a valid response in the opposite direction. For TCP connections, this means a SYN/ACK and for UDP and ICMP traffic, this means a response where source and destination of the original packet are switched.
* RELATED: Packets that are not part of an existing connection, but are associated with a connection already in the system are labeled RELATED. This could mean a helper connection, as is the case with FTP data transmission connections, or it could be ICMP responses to connection attempts by other protocols.
* INVALID: Packets can be marked INVALID if they are not associated with an existing connection and aren’t appropriate for opening a new connection, if they cannot be identified, or if they aren’t routable among other reasons.
* UNTRACKED: Packets can be marked as UNTRACKED if they’ve been targeted in a raw table chain to bypass tracking.
* SNAT: This is a virtual state set when the source address has been altered by NAT operations. This is used by the connection tracking system so that it knows to change the source addresses back in reply packets.
* DNAT: This is a virtual state set when the destination address has been altered by NAT operations. This is used by the connection tracking system so that it knows to change the destination address back when routing reply packets.

General format of commands in iptables is as below



As it is clear after commad iptables we must determine -t ***table\_name***  and then which behavior must be done

-I : insert entry

-A : append entry

-D : delete entry

-U : update an available entry

After that we determine line number which this command must be insert/delete/update

Then -p ***protocol\_type***  and then -s ***source\_ip*** and then -d ***destination\_ip***

Then - -sport ***source\_port\_number*** and then - - dport ***destination\_port\_number***

Then we must determine -m ***module***

Most popular modules :

* + **conntrack**: This is one of the commands made available by calling the conntrack module. This command allows us to match packets**.** We pass it the value of ESTABLISHED to allow packets that are part of an existing connection. We pass it the value of RELATED  
    # iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

At the end main command

1. ***ACCEPT***
2. ***REJECT***
3. ***DROP***
4. ***MASQURADE***
5. ***DNAT***
6. ***SNAT\_***

# Show rules

# To show the tables and commands **: Iptables -nL -t *table-name***

**#** to show rules with line number : **iptables -L --line-numbers**

# Saving Rules

**Debian/Ubuntu**: iptables-save > /etc/iptables/rules.v4  
**RHEL/CentOS**: iptables-save > /etc/sysconfig/iptables

# to restart service of iptables

Systemctl restart iptables.service

# Flush iptables rules

# iptables -F -t *table-name chain-name*

# iptables -F # to flush all rules of iptables